Hand Shape Recognition Using 3D Active Appearance Models

Ryo Yamashita[1], Tetsuya Takiguchi[2], Yasuo Ariki[3]

Graduate School of System Informatics, Kobe University 1-1 Rokkodai, Nada, Kobe, 657-8501, Japan

[1]ryo@me.cs.scitec.kobe-u.ac.jp; [2]takigu@kobe-u.ac.jp; [3]ariki@kobe-u.ac.jp

Abstract

In this paper, a recognition method to discern complicated hand shapes has been proposed using 3D models as an interface for high-functionality TV. In such an interface, a user has to show his hand directly in front of the camera installed on the TV because it cannot recognize the hand shape when viewed in arbitrary directions. With this problem in mind, we have made it possible to track hand shapes in any direction by using 3D active appearance models (3D-AAMs). With the high-functional range image sensor Kinect, RGB images and depth-images of the targets can be obtained; by which hand shape models are constructed. Using multiple 3D AAMs, the robust recognition of such complicated hand shapes in any direction becomes possible.

Keywords

Shape Recognition; Active Appearance Model; 3D Hand Model

Introduction

Recently, new interface techniques meeting high demand for use in combination with high-functionality TV or personal computers that use hand gestures in order to free users up from remote-controls. With [Kinect for Xbox 360], we can use our hands in place of a mouse to control a pointer, or play movies and music. There are many advantages of being free from a remote-control. For example, we can handle the main devises directly with our hands without keeping in mind how to use the controller buttons, or concerning that it runs out of batteries.

However, even if a high-functional range image sensor is in utilization, there will be some limitations in using hand gestures, compared to a conventional remote-control, namely limitations associated with the field of recognition and the ability to recognize finger motions. In the Kinect interface, users control the pointer on the TV display with their hands and keep the pointer still for a while on the icon they want. Then they can convey their commands to the TV. This method is not only time-consuming in regard to giving operation commands, but it also easily results in mistakes in controlling the pointer.

On the other hand, when images are used, taken through a camera, even the finger shape can be recognized by using appearance information, and the total cost will be reduced. However, this method requires user to put his hands in front of a camera and in some cases, this makes things difficult for the user.

With these things in mind, in this paper, a method to recognize gestures has been put forward (including the shape of fingers) using both depth information and appearance information. At first, multiple 3D hand and finger models are constructed using depth information and appearance information. Then the complicated shapes of the hand and fingers are recognized in any direction, and the 3D model is switched according to the shape change.

System Flow

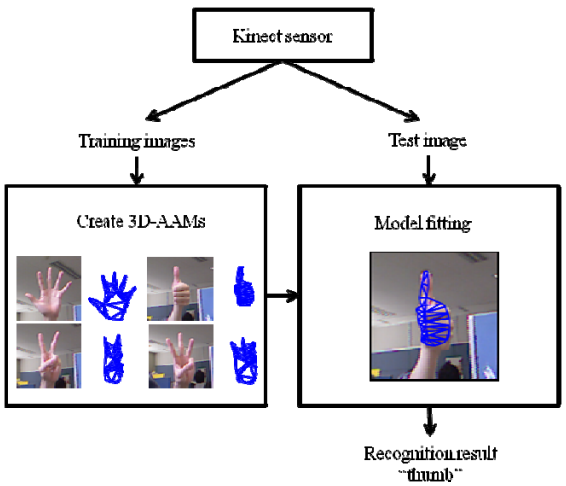

FIG. 1 SYSTEM FLOW

Fig. 1 shows the flow of the proposed system. In the learning phase, multiple 3D models are trained using depth information obtained from Kinect and RGB images. In this study, the 3D-AAM is applied, which combines active appearance models (AAMs) [T.F.

Cootes, 1995] [T.F. Cootes, 1998] [T. F. Cootes, 2002] and three-dimensional information as a three-dimensional model [J. Xiao, 2004] [M. Zhou, 2010] [V. Blanz, 1999]. The next step in the recognition phase is to fit each 3D model to the input images obtained from the Kinect sensor. Finally, the finger shape is output, which has a minimal difference between the model and the input finger.

Hand Feature Extraction Using Multiple 3D-AAMs

Hand feature extraction using multiple 3D-AAMs [J. Xiao, 2004] is described in this section.

Active Appearance Models

Cootes proposed an AAM that is mainly used to extract feature points of a face to represent the shape and texture variations of an object with a low dimensional parameter vector [T.F. Cootes, 1995]. The subspace is constructed by the application of principal component analysis (PCA) to the shape and texture of an object's feature points.

In the AAM framework, shape vector x and texture vector g of the object are given as follows:

$$x = (x_1, y_1, \cdots, x_n, y_n)^T \quad (1)$$

$$g = (g_1, g_2, \cdots, g_n)^T \quad (2)$$

where the shape x indicates the coordinates of the feature points, and the texture vector g indicates the gray-level of the image within the shape. For example, the AAM of the hand is constructed using 44 shape points as shown in Fig. 2. The texture consists of the intensities at pixels within triangular areas with feature points as shown in Fig. 3.

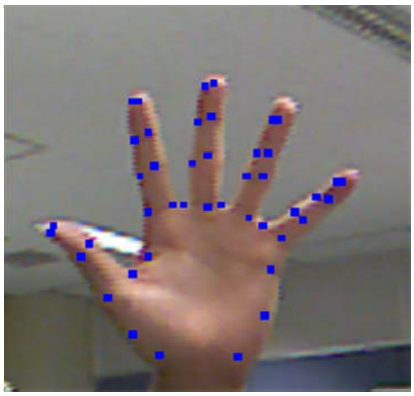

FIG. 2 HAND FEATURE POINTS

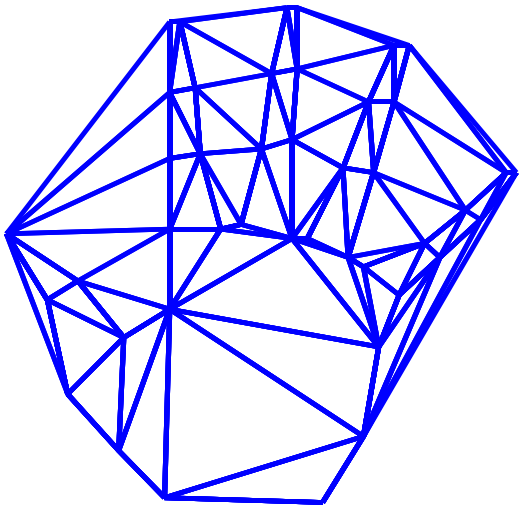

FIG. 3 TRIANGULAR AREAS IN SHAPE

Next, PCA is applied to the training data in order to obtain the normal orthogonal matrices, P_S and P_g. Using the obtained matrices, the shape vector and texture vector can be approximated as follows:

$$x = \bar{x} + P_s b_s \quad (3)$$

$$g = \bar{g} + P_g b_g \quad (4)$$

where $\bar{x}$ and $\bar{g}$ are the mean shape and mean texture of the training images, respectively. b_s and b_g are the parameters of variation from the average. Further PCA is applied to the vector b as follows:

$$b = \begin{pmatrix} W_s b_s \\ b_g \end{pmatrix} + \begin{pmatrix} W_s P_s^T (x - \bar{x}) \\ P_g^T (g - \bar{g}) \end{pmatrix} = Qc \quad (5)$$

$$Q = \begin{pmatrix} Q_s \\ Q_g \end{pmatrix} \quad (6)$$

where W_s is a diagonal weight matrix for each shape parameter, allowing for the difference in units between the shape and texture models. Q_s and Q_g are the eigen matrices (including the eigenvectors). c is a vector of parameters controlling both the shape and gray-levels of the model. Finally, the shape and texture are approximated as functions of c.

$$x(c) = \bar{x} + P_s W_s^{-1} Q_s c \quad (7)$$

$$g(c) = \bar{g} + P_g Q_g c \quad (8)$$

Pose Parameter

Using parameter c, it is possible to control variations in the shape and texture of the AAM. However, it is

not possible to express the position of the object in the image, the size of the object, or the object pose. The pose parameter q is defined as the global posture change as follows:

$$q = [roll \quad scale \quad trans\_x \quad trans\_y] \tag{9}$$

where $roll$ indicates the rotation to the model plane, $scale$ indicates the size of the model, while $trans\_x$ and $trans\_y$ indicate the translation between x and y, respectively.

Tracking AAM

The goal of the AAM search is to minimize the error E on the test image *Img* as shown in Eq. (10) with respect to c and q,

$$E = \left[\left(\bar{g} + P_g Q_g c'\right) - I\left(Img, W\left(x; q', b_s'\right)\right)\right]^2 \tag{10}$$

where W denotes the Affine warp function, and $I\left(Img, W\left(x; q', b_s'\right)\right)$ indicates the Affine transformed image controlled by the pose parameter q on the test image. Thus, the most optimized c parameter can be extracted from the test image.

3D-AAM

In this paper, the 3D-AAM is used to extract the hand feature and to estimate the shape of the fingers. The AAM includes various fluctuation components (images) because the object images used as the training data include various changes, such as a left-side object image or right-side one, and an upturned or a downturned object. However, if so many changes are included in the training data of the object, the AAM cannot express their variation in PCA. Then the extraction accuracy of feature points will become lower. Since the variation of directions can be expressed as geometric change of shape, if a 3D shape with a right texture can be used, it can express the directional variations. This is the reason why the 3D-AAM is employed in the hand and finger shape recognition. The shape parameter is expanded into 3D using z obtained from a depth image sensor, as shown in Eq. (11).

$$x = \left(x_1, y_1, z_1, \cdots, x_n, y_n, z_n\right)^T \tag{11}$$

RGBD Images Obtained Using Depth Image Sensor

In the course of the creation of the 3D-AAM, three-dimensional shape information is required for a target. The 3D shape can be obtained, for example, using a 3D scanner or a stereo camera. However, we chose to use a Kinect sensor because this device is equipped with an RGB camera and infrared depth sensor. The 3D data expressed in Eq. (11) is obtained as a set of coordinate points on a target using a Kinect depth image sensor.

Elimination of Background Images in Triangular Areas Using Background Subtraction

Background images are included in triangular areas of Fig. 3. Therefore, in order to eliminate the background images, background subtraction is performed using the depth data obtained from depth image sensor (Fig. 4).

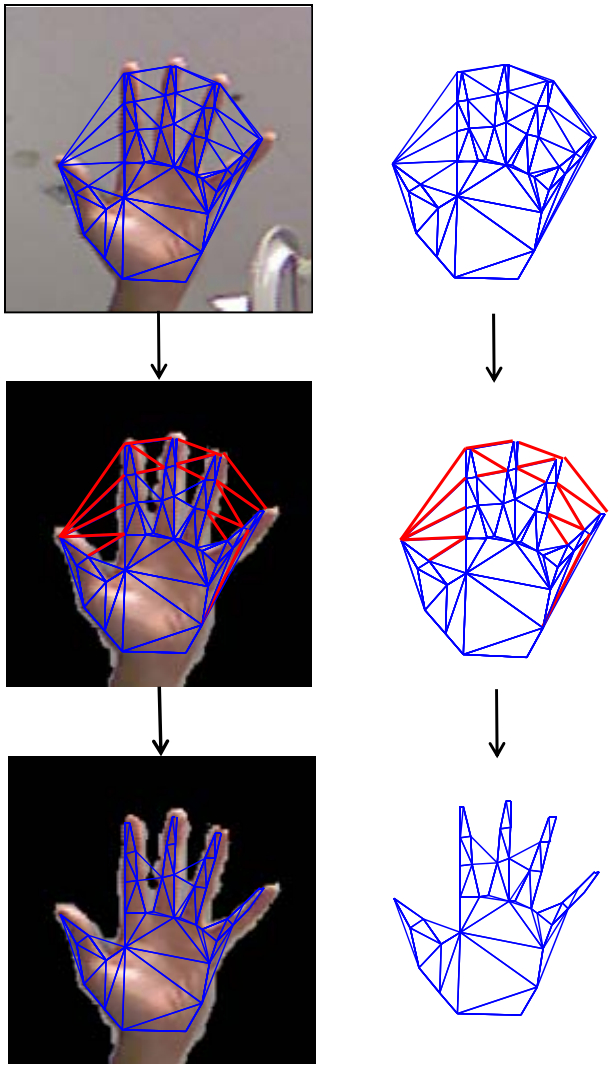

FIG. 4 BACKGROUND SUBTRACTION

Expanding of Pose Parameters

The 2D pose parameters in Eq. (9) are expanded into 3D by adding yaw and pitch as shown in Eq. (12).

$$q = [yaw \quad pitch \quad roll \quad scale \quad trans\_x \quad trans\_y] \tag{12}$$

The moving variations of these parameters are shown

in Fig. 5.

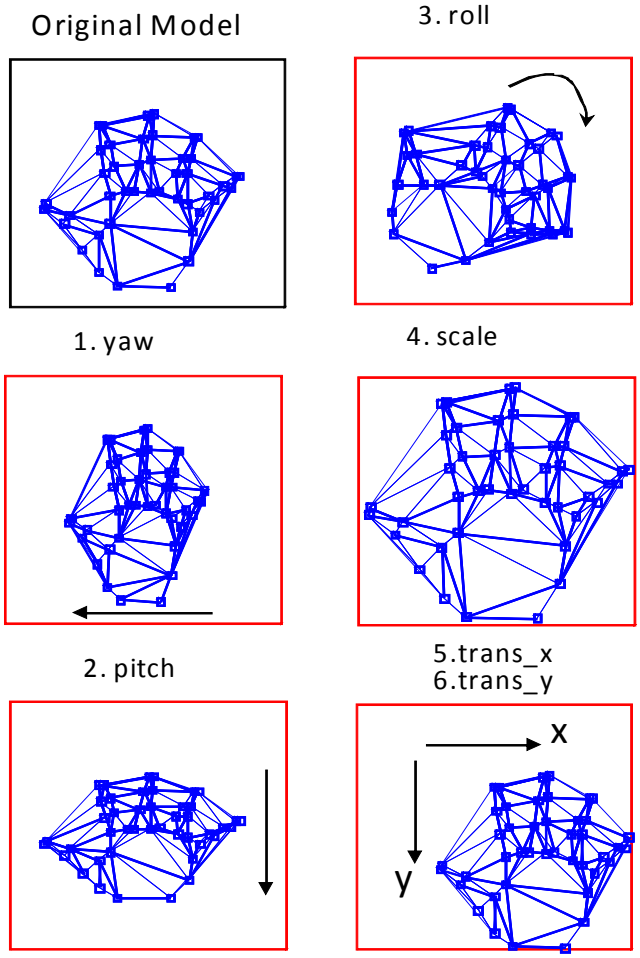

FIG. 5 3D POSE PARAMETERS

Using the six parameters, the 2D-AAM can be expanded into the three-dimensional AAM, and the parameters can transform the model viewed in all directions, angles, and positions. The transformation of the shape using this pose parameter is given as follows:

$$x_a = trans \cdot Scale \cdot RotZ \cdot RotY \cdot RotX \cdot x_b \qquad (13)$$

where x_a and x_b indicate the shape coordinate after and before transformation, respectively. Each transformation matrix is given by Eq. (14)-(18).

$$Trans = \begin{pmatrix} 1 & 0 & 0 & trans\_x \\ 0 & 1 & 0 & trans\_y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad (14)$$

$$Scale = \begin{pmatrix} scale & 0 & 0 & 0 \\ 0 & scale & 0 & 0 \\ 0 & 0 & scale & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad (15)$$

$$RotX = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(yaw*\pi/180) & -\sin(yaw*\pi/180) & 0 \\ 0 & \sin(yaw*\pi/180) & \cos(yaw*\pi/180) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad (16)$$

$$RotY = \begin{pmatrix} \cos(pitch*\pi/180) & 0 & \sin(pitch*\pi/180) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(pitch*\pi/180) & 0 & \cos(pitch*\pi/180) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad (17)$$

$$RotZ = \begin{pmatrix} \cos(roll*\pi/180) & -\sin(roll*\pi/180) & 0 & 0 \\ \sin(roll*\pi/180) & \cos(roll*\pi/180) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad (18)$$

Tracking the 3D-AAM

Since the 3D-AAM consists of three-dimensional points and the input image consists of two dimensional pixels, it is necessary to project the 3D space into the 2D space when the error between the input image and the 3D model is calculated. Using the function P which projects the 3D space into the 2D space, the error can be calculated by Eq. (19). Then the optimized parameters are calculated using the same approach used in the 2D-AAM.

$$E = \left[\left(\bar{g} + P_g Q_g c'\right) - I\left(\mathrm{Im}\, g, P\left(W\left(x; q', b_s'\right)\right)\right)\right]^2 \qquad (19)$$

Experiments

Multiple 3D-AAMs are constructed for the finger shapes and the model that produces the smallest error when the data is input into Eq. (19) is selected as the final result, as a sequence in the finger movie.

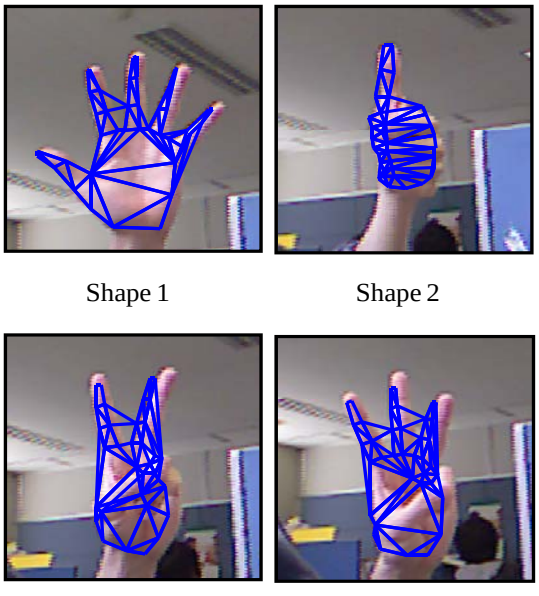

FIG. 6 FOUR MODELS OF FINGER SHAPE

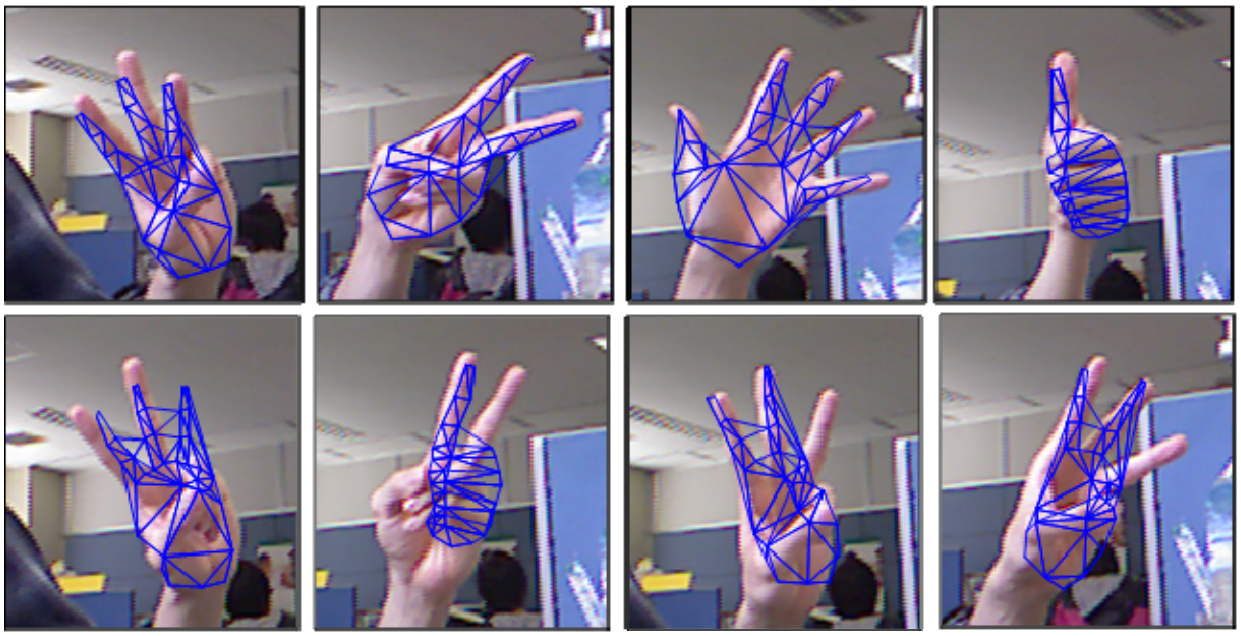

FIG 7. EXAMPLES OF EXPERIMENT RESULTS

Experiment Conditions

Four types of three-dimensional models shown in Fig. 6 have been constructed in this experiment. The number of feature points included in respective model is 44, 34, 34 and 38 points, respectively. During the construction of each model, four depth images have been used to learn. Before learning, the background included in the training images has been excluded based on the background-subtraction. The training data and testing data collected from the same single person; included the finger shape with several variations taken from different directions.

Experiment Results

Table 1 shows the confusion matrix of the finger shape recognition results. As a result of the experiment, it showed that the higher recognition rate has been obtained for shape 1 and shape 2. However, the false recognition can be seen between shape 3 and shape 4. It is thought that their finger shapes are similar so that the error of Eq. (19) becomes smaller. Fig. 7 shows the example of the outcome of this experiment.

TABLE 1 CONFUSION MATRIX AMONG FOUR SHAPE MODELS

	Shape 1	Shape 2	Shape 3	Shape 4
Shape 1	1.0	0.0	0.0	0.0
Shape 2	0.0	1.0	0.0	0..0
Shape 3	0.0	0.011	0.966	0023
Shape 4	0.0	0.0	0.327	0.673

Conclusions

In this paper, by means of multiple three-dimensional models, the finger shape recognition method has been proposed. Using depth image sensor as a device, three-dimensional models were constructed by the acquisition of the depth information and RGB images of objects. The model recognized the finger shapes robustly against the various changes. The improvement in the finger shape models to be applied in the interface to TV or other display devices is the focus of further research.

REFERENCES

Jing Xiao, Simon Baker, Iain Matthews, and Takeo Kanade, "Real-Time Combined 2D+3D Active Appearance Models." CVPR, 535-542, 2004.

Mingcai Zhou, Yangsheng Wang, and Xiangsheng Huang, "Real-Time 3D Face and Facial Action Tracking Using Extended 2D+3D AAMs." ICPR, 3963-3966, 2010.

T.F. Cootes, C.J. Taylor, D.H. Cooper, and J. Graham, "Active Shape Models – Their Training and Applications." Computer Vision and Image Understanding, Vol. 6, No. 1, 38-59, 1995.

T.F. Cootes, G.J. Edwards, and C.J. Taylor, "Active appearance models." ECCV, 484-498, 1998.

T.F. Cootes, K. Walker, and C.J. Taylor, "View-Based Active Appearance Models." Image and Vision Computing, 657-664, 2002.

V. Blanz and T. Vetter, "A morphable model for the synthesis of 3D faces." SIGGRAPH, 187-194, 1999.

Ryo Yamashita received the B.E. in computer science from Kobe University in 2011. He is currently in graduate school of system informatics, Kobe University.

Tetsuya Takiguchi received the B.S. degree in applied mathematics from Okayama University of Science, Okayama, Japan, in 1994, and the M.E. and Dr. Eng. degrees in information science from Nara Institute of Science and Technology, Nara, Japan, in 1996 and 1999, respectively.

From 1999 to 2004, he was a researcher at IBM research, Tokyo Research Laboratory, Kanagawa, Japan. Since 2004 he has been an Associate Professor at Kobe University. He stayed at University of Washington as visiting scholar from April 2012 to October 2012.

Dr. Takiguchi is a member of IEEE, Information Processing Society of Japan, and Acoustical Society of Japan.

Yasuo Ariki received his B.E., M.E. and Ph.D. in information science from Kyoto University in 1974, 1976 and 1979, respectively.

He was an assistant professor at Kyoto University from 1980 to 1990, and stayed at Edinburgh University as visiting academic from 1987 to 1990. From 1990 to 1992 he was an associate professor and from 1992 to 2003 a professor at Ryukoku University. Since 2003 he has been a professor at Kobe University. He is mainly engaged in speech and image recognition and interested in information retrieval and database.

Prof. Ariki is a member of IEEE, IPSJ, JSAI, ITE and IIEEJ.